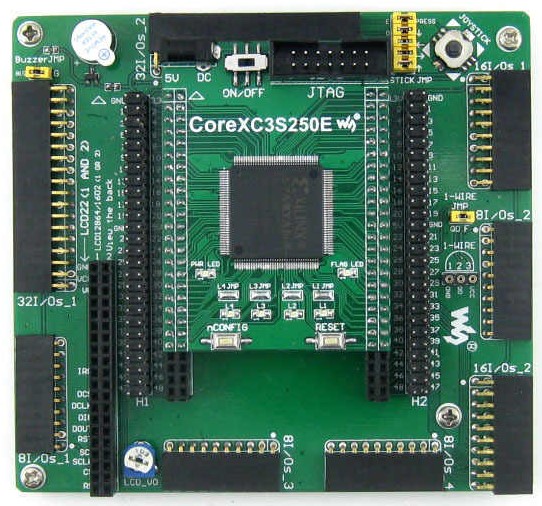
|  |  |  |  |
| --- | --- | --- | --- |
| Nome: | **Pedro Gabriel G. R. Balestra** | Curso: **GEC** | Matrícula: **1551** |
| Nome: | **Victor Alexandre P. Marques** | Curso: **GET** | Matrícula: **9629** |
| Nome: | **Wesley Marcos Borges** | Curso: **GEC** | Matrícula: **1651** |
| **INATEL – 2º SEMESTRE - 2021** | | Matéria: **E208** | Turma: **L13** |

**Projeto do Laboratório de Eletrônica Digital 2**

1. **Pesquisa**
   1. **O que é Spartan 3E**

O Spartan 3E é uma ferramenta de desenvolvimento autônoma FPGA (Field Programable Gate Arrays) que utiliza da linguagem VHDL. Com essa ferramenta, é possível realizar diversas ações como conversão de sinal analógico para digital, ou vice-versa, implementação de circuitos lógicos digitais programáveis, armazenamento de dados em memória RAM dinâmica ou memória flash e, dentre outras.

**Placa Spartan 3E**

* 1. **Como funciona uma placa Spartan 3E**

Como dito anteriormente, deu funcionamento se dá através da linguagem VHDL ou Verilog. Com essa placa, é possível testar de maneira física o que é visto na simulação através do software Xilinx ISE, em todas as versões do software, já que o VHDL nos dá acesso aos componentes necessários para uso dessa placa. Ela é fornecida junto a uma fonte de alimentação e um cabo USB para a implementação dos projetos. Outra forma de acessar os componentes é através do diagrama de componentes, onde a descrição é encapsulada e apresentada em um bloco de portas IN/OUT, permitindo o funcionamento.

* 1. **Onde, como e com qual finalizade essa placa é usada**

Essa placa é bastante usada na prototipagem de dispositivos na produção em baixa escala, no âmbito industrial. Normalmente, ela é acoplada à circuitos que usam portas lógicas, pois sua implementação auxilia no menor custo do projeto e na diminuição do número de processadores e máquinas digitais.

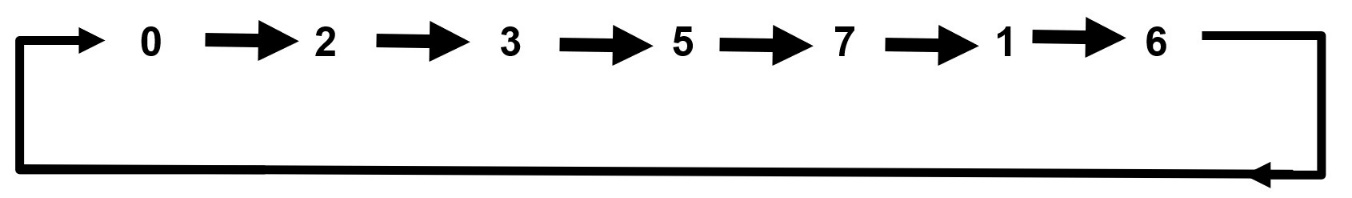
* 1. **A importância de se aprender a linguagem de programaçãp VHDL**

A linguagem VDHL é uma linguagem de descrição de hardware, criada à partir da necessidade de uma ferramenta pudesse simular, descrever e documentar projetos no padrão VHSIC (Very High Speed Integrated Circuit), através de níveis hierárquicos. A possibilidade de se descrever a operação lógica interna de um bloco, por exemplo, usando enunciados bem definidos, tais como uma linguagem de programação convencional, faz com que o requinte da linguagem seja ainda maior.

Devido a sua grande necessidade e aceitação, aprender a linguagem VHDL é de suma importância, devido às suas características essenciais no desenvolvimento e prototipagem, visto que ela auxilia de diversas maneiras, evitando, por exemplo, gastos com testes físicos, elevando o nível de qualidade dos circuitos.

1. **Condições Iniciais do Projeto**
   1. **Definição do Número de Flip Flops**

Nosso projeto é baseado na seguinte contagem síncrona:



Como o maior número é o , usaremos 3 Flip Flops, pois = 111.

1. **Montando o contador síncrono** 
   1. **Usando o FF – D**

Para montar a tabela necessária, usaremos a tabela do FF – D como base:

|  |  |  |
| --- | --- | --- |
| Qa | Qf | D |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Assim sendo, a tabela verdade montada com FF – D ficou da seguinte forma:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qa (Estado Anterior) | | | Qf (Estado Final) | | | FF - D | | |
| Q2 | Q1 | Q0 | Q2 | Q1 | Q0 | D2 | D1 | D0 |
| **0** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **0** |
| **0** | **0** | **1** | **1** | **1** | **0** | **1** | **1** | **0** |
| **0** | **1** | **0** | **0** | **1** | **1** | **0** | **1** | **1** |
| **0** | **1** | **1** | **1** | **0** | **1** | **1** | **0** | **1** |
| **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **1** |

Simplificando as expressões obtidas em D2, D1 e D0, obtemos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D2 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | 0 | 0 | 0 | 0 |
| Q0 | 1 | 1 | 0 | 1 |

**D2 = Q0. + Q0.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D1 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | 1 | 0 | 0 | 1 |
| Q0 | 1 | 1 | 0 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D0 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | 0 | 0 | 0 | 1 |
| Q0 | 0 | 1 | 1 | 1 |

**D1 = . + Q0.  
  
  
  
  
  
  
  
D0 = Q0.Q1 + Q1.**

* 1. **Usando o FF – T**

Para montar a tabela necessária, usaremos a tabela do FF – T como base:

|  |  |  |
| --- | --- | --- |
| Qa | Qf | T |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Assim sendo, a tabela verdade montada com FF – T ficou da seguinte forma:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qa (Estado Anterior) | | | Qf (Estado Final) | | | FF - T | | |
| Q2 | Q1 | Q0 | Q2 | Q1 | Q0 | T2 | T1 | T0 |
| **0** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **0** |
| **0** | **0** | **1** | **1** | **1** | **0** | **1** | **1** | **1** |
| **0** | **1** | **0** | **0** | **1** | **1** | **0** | **0** | **1** |
| **0** | **1** | **1** | **1** | **0** | **1** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **1** | **0** | **1** | **1** | **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** | **0** | **0** | **1** | **1** | **0** |
| **1** | **1** | **1** | **0** | **0** | **1** | **1** | **1** | **0** |

Simplificando as expressões obtidas em T2, T1 e T0, obtemos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T2 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | 0 | 1 | 1 | 0 |
| Q0 | 1 | 0 | 1 | 1 |

**T2 = .Q2 + Q0.Q1 + Q0.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T1 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | 1 | 0 | 1 | 0 |
| Q0 | 1 | 1 | 1 | 1 |

**T1 = . + Q1.Q2 + Q0**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T0 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | 0 | 0 | 0 | 1 |
| Q0 | 1 | 0 | 0 | 0 |

**T0 = Q0.. + Q1.**

* 1. **Usando o FF - JK**

Para montar a tabela necessária, usaremos a tabela do FF – JK como base:

|  |  |  |  |
| --- | --- | --- | --- |
| Qa | Qf | J | K |
| 0 | 0 | 0 | \* |
| 0 | 1 | 1 | \* |
| 1 | 0 | \* | 1 |
| 1 | 1 | \* | 0 |

Assim sendo, a tabela verdade montada com FF – T ficou da seguinte forma:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qa (Estado Anterior) | | | Qf (Estado Final) | | | FF-JK | | | | | |
| Q2 | Q1 | Q0 | Q2 | Q1 | Q0 | J2 | K2 | J1 | K1 | J0 | K0 |
| **0** | **0** | **0** | **0** | **1** | **0** | **0** | **\*** | **1** | **\*** | **0** | **\*** |
| **0** | **0** | **1** | **1** | **1** | **0** | **1** | **\*** | **1** | **\*** | **\*** | **1** |
| **0** | **1** | **0** | **0** | **1** | **1** | **0** | **\*** | **\*** | **0** | **1** | **\*** |
| **0** | **1** | **1** | **1** | **0** | **1** | **1** | **\*** | **\*** | **1** | **\*** | **0** |
| **1** | **0** | **0** | **0** | **0** | **0** | **\*** | **1** | **0** | **\*** | **0** | **\*** |
| **1** | **0** | **1** | **1** | **1** | **1** | **\*** | **0** | **1** | **\*** | **\*** | **0** |
| **1** | **1** | **0** | **0** | **0** | **0** | **\*** | **1** | **\*** | **1** | **0** | **\*** |
| **1** | **1** | **1** | **0** | **0** | **1** | **\*** | **1** | **\*** | **1** | **\*** | **0** |

Simplificando as expressões obtidas em J2, J1, J0, K2, K1 e K0, obtemos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| J2 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | 0 | \* | \* | 0 |
| Q0 | 1 | \* | \* | 1 |

**J2 = Q0**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K2 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | \* | 1 | 1 | \* |
| Q0 | \* | 0 | 1 | \* |

**K2 = + Q1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| J1 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | 1 | 0 | \* | \* |
| Q0 | 1 | 1 | \* | \* |

**J1 = Q0 + 𝐐𝟐**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K1 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | \* | \* | 1 | 0 |
| Q0 | \* | \* | 1 | 1 |

**K1 = Q0 + Q2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| J0 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | 0 | 0 | 0 | 1 |
| Q0 | \* | \* | \* | \* |

**J0 =**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K0 | Q1'Q2' | Q1'Q2 | Q1Q2 | Q1Q2' |
| Q0' | \* | \* | \* | \* |
| Q0 | 1 | 0 | 0 | 0 |

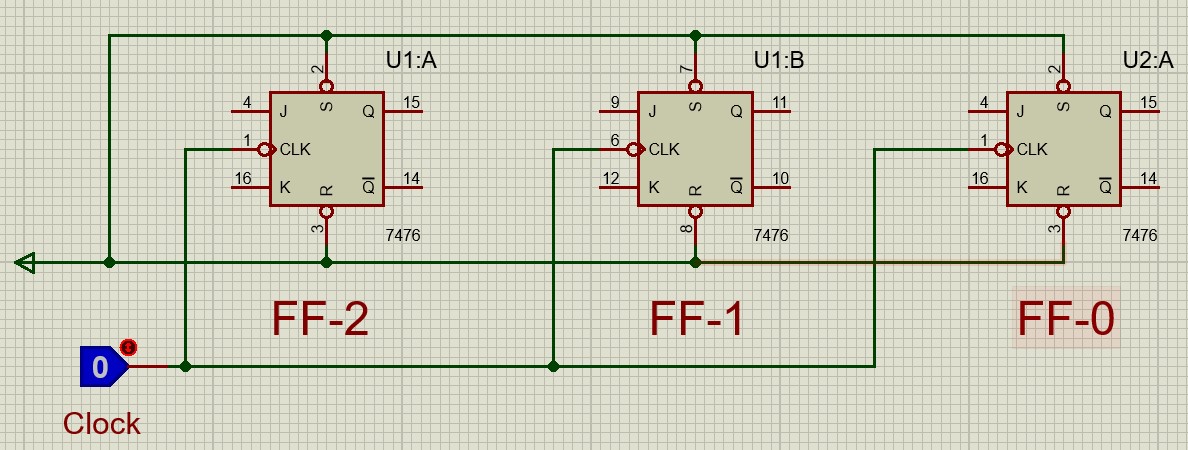
**K0 = .**

Partindo das simplificações feitas nos FF’s JK, D e T, **escolhemos o JK**, pois é o mais simples de ser implementado.

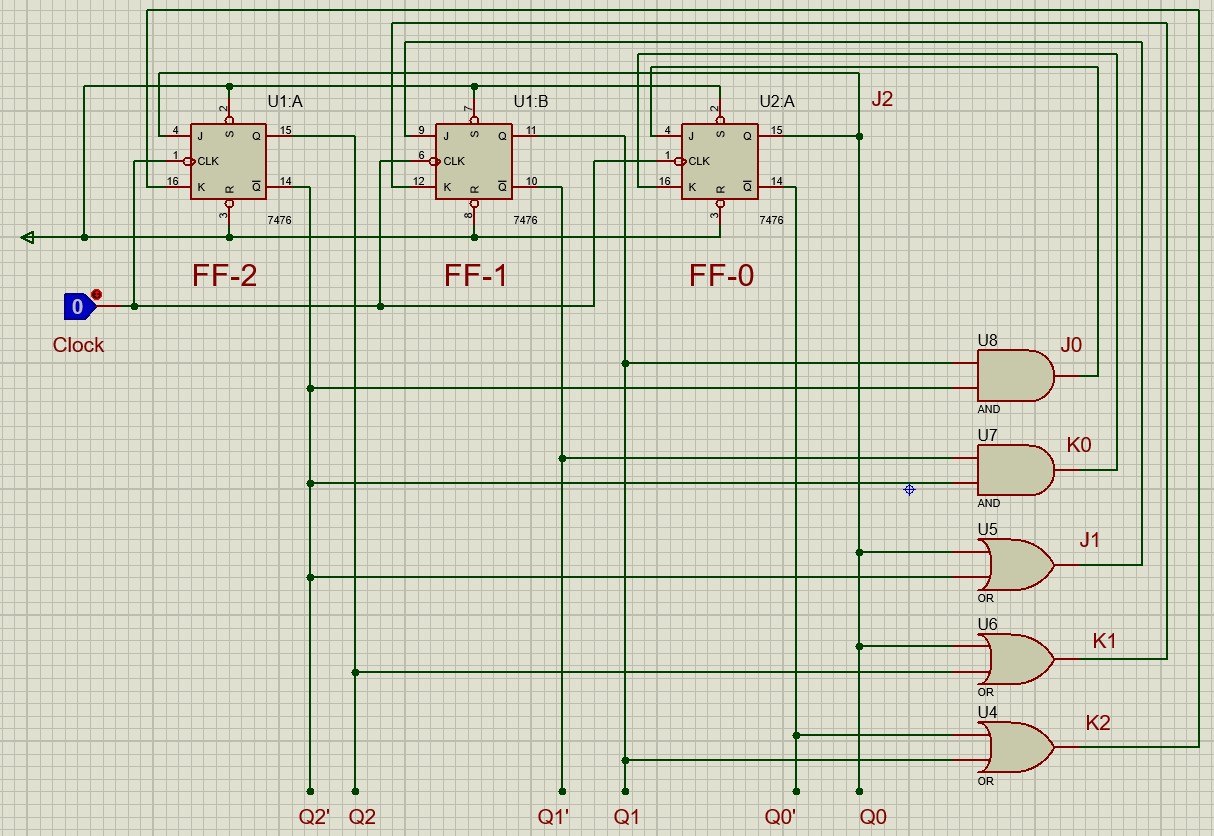
1. **Simulação do contador síncrono no Proteus**

Definimos a montagem no Proteus através de 3 passos:

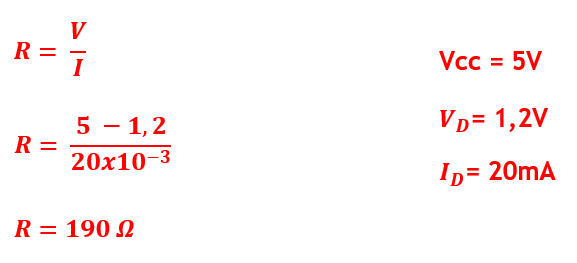
* 1. Primeiro, ligamos os clock’s em série, conectando-os no mesmo terminal. Também desacionamos o set e preset, ligando-os no VCC.



* 1. Segundo, conectamos ao circuito as entradas J e K dos respectivos Flip-Flops, usando portas AND e OR, como definido anteriormente.



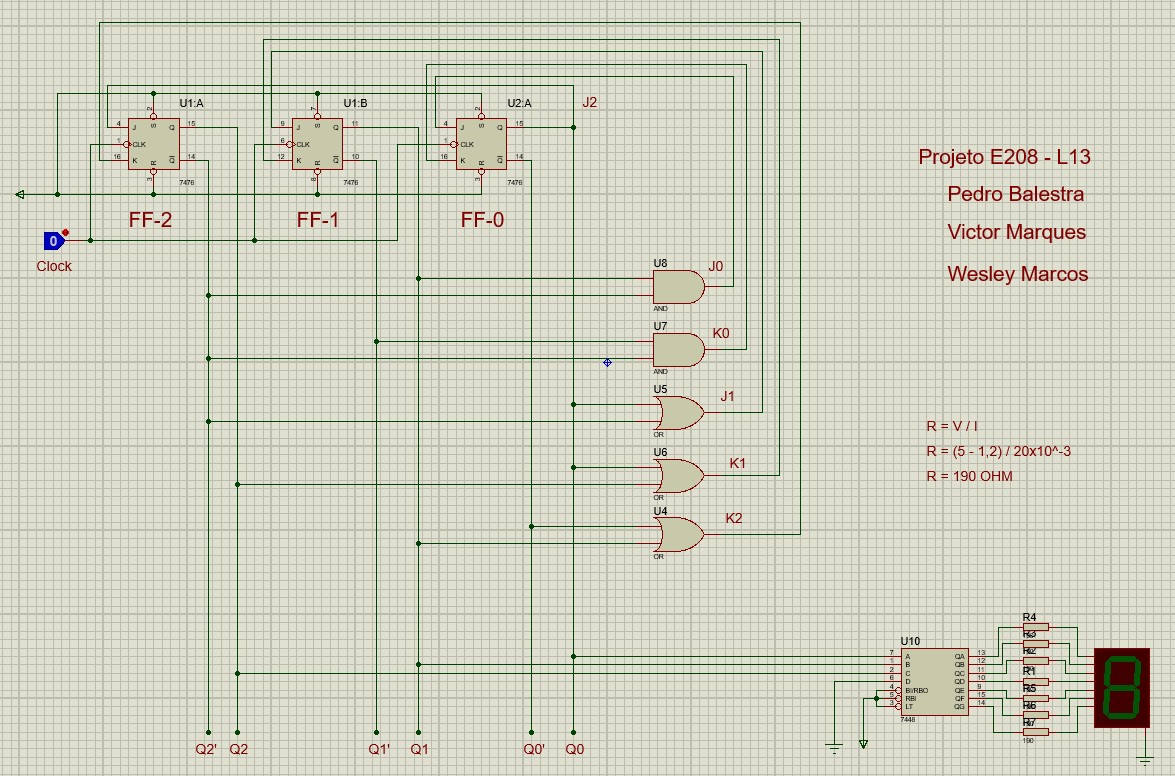
* 1. Terceiro, conectamos ao circuito um decodificador 7448, responsável por fazer a conversão dos números binários para decimais, através das entradas A, B e C. Nas saídas, conectamos 7 resistores, calculados da seguinte forma:



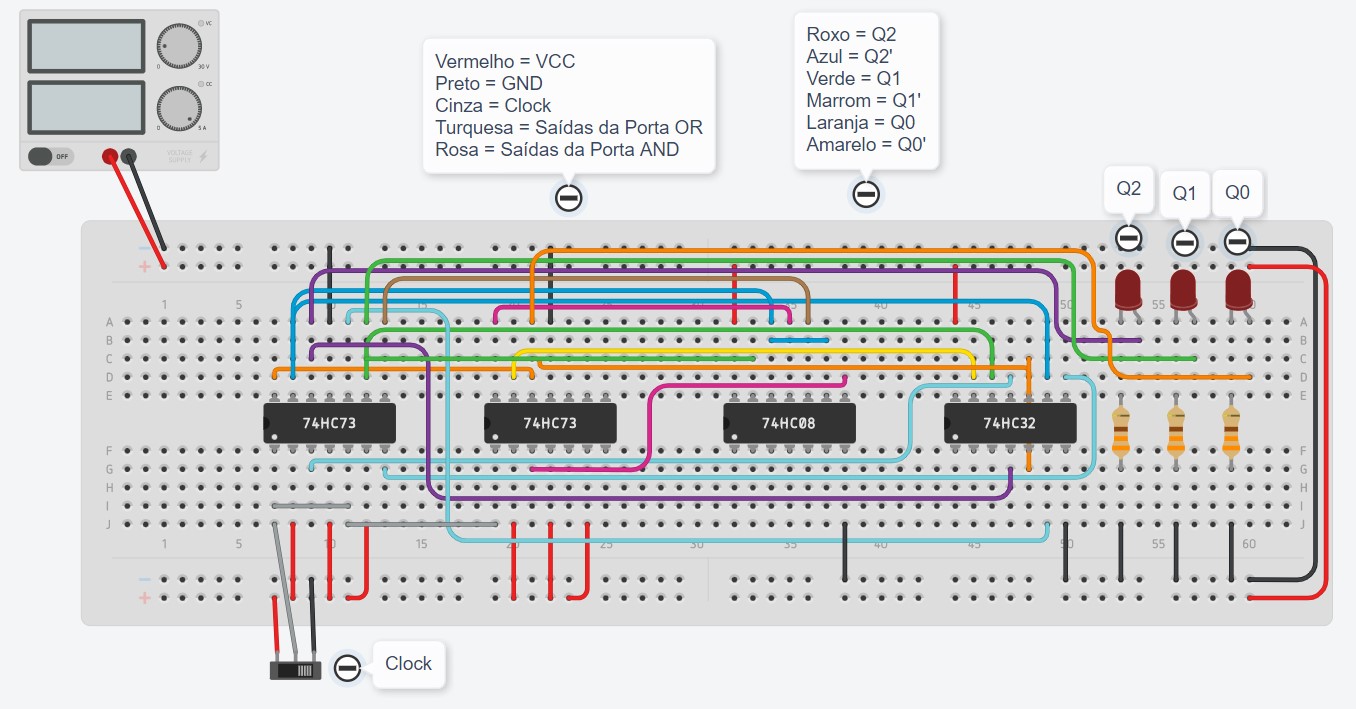
Após os resistores, conectamos um display 7 segmentos com catodo comum, terminando assim a conversão para números decimais.



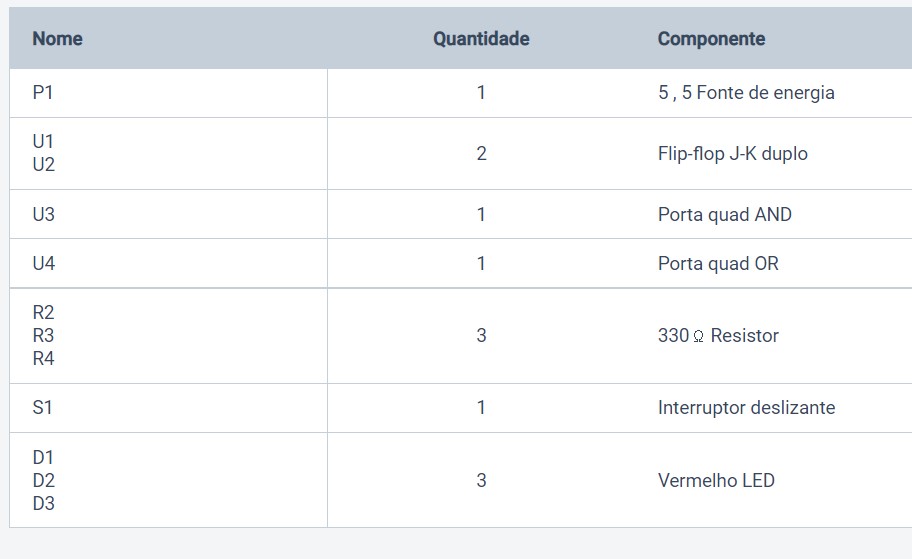
* 1. O circuito completo ficou assim:



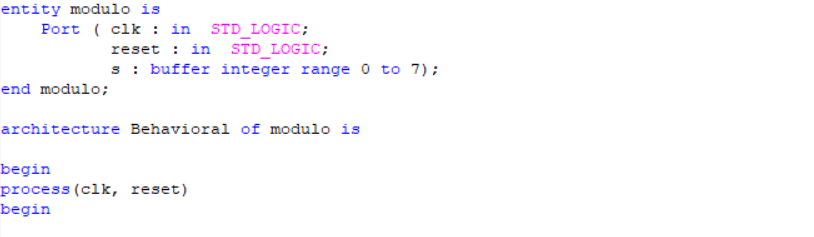
1. **Simulação do contador síncrono no Tinkercad**
   1. **Circuito Elétrico**

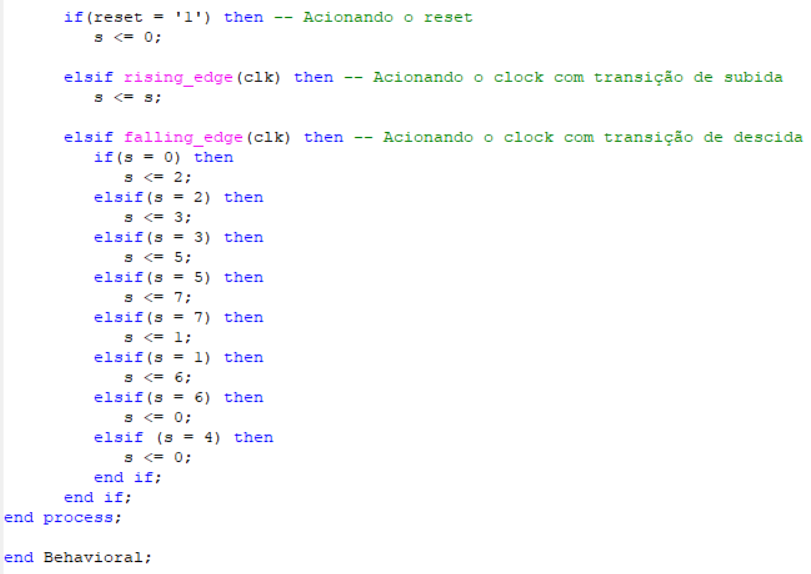


* 1. **Tabela de Componentes**

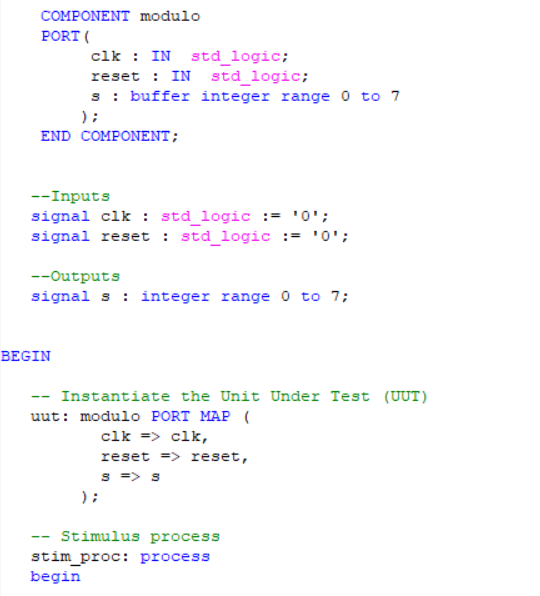


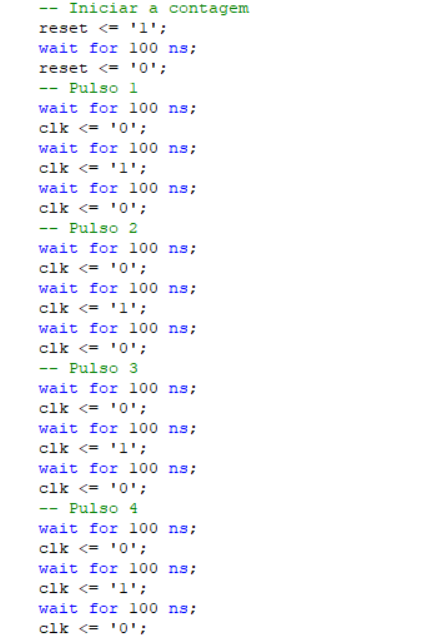
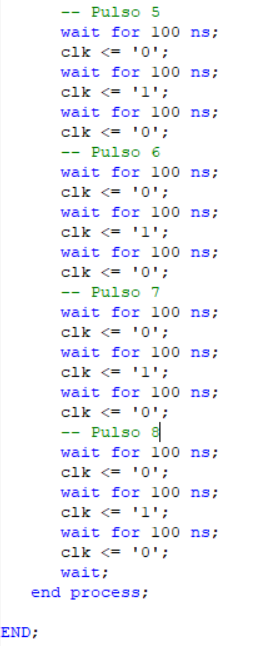
1. **Simulação do contador síncrono no ISE**
   1. **Módulo**

****

****

* 1. **Teste**

****

****

* 1. **Simulação**

